

Задача А. Ретроанализ для маленьких

Имя входного файла: `retro.in`
Имя выходного файла: `retro.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан ориентированный весёлый граф из n вершин и m ребер. Оля и Коля в игру. Изначально фишка стоит в вершине i . За ход можно передвинуть фишку по любому из исходящих ребер. Тот, кто не может сделать ход, проигрывает. Ваша задача — для каждой вершины i определить, кто выиграет при оптимальной игре обоих.

Формат входных данных

Входные данные состоят из одного или нескольких тестов. Каждый тест содержит описание весёлого ориентированного графа. Граф описывается так: на первой строке два целых числа n ($1 \leq n \leq 300\,000$) и m ($1 \leq m \leq 300\,000$). Следующие m строк содержат ребра графа, каждое описывается парой целых чисел от 1 до n . Пара $a\ b$ обозначает, что ребро ведет из вершины a в вершину b . В графе могут быть петли, могут быть кратные ребра. Сумма n по всем тестам не превосходит 300 000, сумма m по всем тестам также не превосходит 300 000.

Формат выходных данных

Для каждого теста выведите для каждой вершины `FIRST`, `SECOND` или `DRAW` в зависимости от того, кто выиграет при оптимальной игре из этой вершины. Ответы к тестам разделяйте пустой строкой.

Примеры

retro.in	retro.out
5 5	DRAW
1 2	DRAW
2 3	DRAW
3 1	FIRST
1 4	SECOND
4 5	FIRST
2 1	SECOND
1 2	FIRST
4 4	FIRST
1 2	SECOND
2 3	SECOND
3 1	
1 4	

Задача В. Функция Гранди

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан ориентированный ациклический граф. Посчитайте функцию Гранди для каждой стартовой вершины.

Формат входных данных

На первой строке будут даны числа n и m — количество вершин и рёбер в графе ($1 \leq n, m \leq 100\,000$). На следующих m строках содержится по два числа x и y ($1 \leq x, y \leq n$).

Учтите, что в графе могут быть кратные рёбра.

Формат выходных данных

Выведите n чисел — значение функции Гранди для каждой стартовой вершины.

Примеры

stdin	stdout
3 3	2
1 2	1
2 3	0
1 3	
2 1	0
2 1	1

Задача С. Дровосек

Имя входного файла: `woodcut.in`
Имя выходного файла: `woodcut.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Двое играют в следующую игру: имеется дерево с отмеченной вершиной (корнем). Игроки ходят по очереди. За ход игрок разрубает ветку (стирает ребро), причем из двух получившихся компонент связности остается только та, которая содержит корень — остальная отваливается и больше в игре не участвует. Проигрывает тот, кто не может сделать ход.

Определите, может ли выиграть первый игрок, и если да, то укажите любой из его выигрышных ходов.

Формат входных данных

В первой строке входного файла находится 2 числа N и R — количество вершин дерева и номер корня ($2 \leq N \leq 1000\,000$, $1 \leq R \leq N$). Далее следует $N - 1$ строк, в каждой из которых находятся два числа — номера вершин, которые соединяет очередное ребро.

Формат выходных данных

Выведите в выходной файл одно число: 1 или 2 — номер игрока, который выигрывает при правильной игре. Если выигрывает первый игрок, то выведите также любой его выигрышный ход, т.е. порядковый номер ребра во входном файле, которое ему достаточно разрубить первым ходом (число от 1 до $N - 1$).

Примеры

<code>woodcut.in</code>	<code>woodcut.out</code>
5 5	1
2 3	1
1 3	
2 5	
4 5	

Задача D. Битва за кольцо

Имя входного файла: `rings.in`
Имя выходного файла: `rings.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Саруман Белый и Гэндальф Серый решили сыграть в игру. Победителю достается Кольцо Всевластия. Перед игроками лежат кольца, соединенные в K цепочек. Для каждого кольца известно содержание золота в нем в процентах — целое число от 1 до 100. Ходят по очереди. За ход разрешается выбрать одну из цепочек и какое-то кольцо из этой цепочки и дематериализовать все кольца из данной цепочки с процентным содержанием золота не больше, чем у выбранного. При этом, понятно, цепочка может распасться на несколько. Игра продолжается на оставшихся цепочках. Тот, кто дематериализовал последнее кольцо, выиграл. Первым ходит Гэндальф. Определите, может ли Гэндальф выиграть и, если может, какой первый ход он должен для этого сделать.

Формат входных данных

В первой строке дано целое число K ($1 \leq K \leq 50$). В следующих K строках приведены описания цепочек в следующем формате: сперва дана длина цепочки — целое число от 1 до 100, затем — процентные содержания золота в кольцах цепочки. Числа в строке разделены пробелом.

Формат выходных данных

Выведите "S", если Кольцо Всевластия достанется Саруману. В противном случае выведите в первой строке "G", а во второй пару чисел, описывающих выигрышный первый ход Гэндальфа — номер цепочки и номер кольца в ней. Цепочки и кольца внутри цепочек нумеруются с 1. Если существует несколько выигрышных первых ходов, выведите ход с наименьшим номером цепочки, если и таких несколько — с наименьшим номером кольца.

Примеры

<code>rings.in</code>	<code>rings.out</code>
2 3 1 2 1 1 1	G 1 1
2 3 2 1 2 1 1	S

Задача Е. Огромный ним

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Петя и Вася играют в ним, но не простой, а просто огромный. У них есть очень много кучек камней. Кучки разделены на n групп. Группа i состоит из кучек размеров от l_i до r_i включительно. Помогите ребятам понять, кто выиграет при оптимальной игре

Формат входных данных

Первая строка входного файла содержит число n ($1 \leq n \leq 10^5$), следующие n строк содержат пары чисел l_i, r_i ($1 \leq l_i \leq r_i \leq 10^{18}$).

Формат выходных данных

Если первый игрок проигрывает, выведите **Lose**, если выигрывает — выведите в первой строке **Win**, а во второй строке — любой выигрышный ход для первого игрока. Ход задается размером кучки до хода и после него.

Примеры

стандартный ввод	стандартный вывод
1 1 10	Win 9 2
2 2 5 2 5	Lose

Задача F. Игра на графе

Имя входного файла: `game.in`
Имя выходного файла: `game.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Геннадий и Георгий играют в интересную игру на ориентированном графе. В графе n вершин и m рёбер, возможны петли. Перед началом игры Геннадий и Георгий поставили в какую-то вершину фишку. Игроки ходят по очереди, перемещая фишку вдоль какого-то ребра, исходящего из вершины, в которой она находится. Если у игрока нет ни одного хода, он проигрывает.

Ваша задача — для каждой вершины графа и для обоих игроков определить результат игры, если фишка стоит в этой вершине и если этот игрок ходит первым. Звучит просто? Всё не совсем так.

С одной стороны, Геннадий получает удовольствие от процесса игры, и хочет, чтобы игра длилась как можно дольше. Он даже предпочтёт бесконечную игру победе. Но если он не может играть бесконечно, он, очевидно, предпочтёт поражению победу.

С другой стороны, у Георгия куча дел, и он не хочет играть бесконечно. Георгий хочет победить, но если он не может, он предпочтёт поражение бесконечной игре.

Оба игрока играют оптимально. Оба игрока знают предпочтения противника.

Формат входных данных

В первой строке записано два целых числа — число вершин n ($1 \leq n \leq 100\,000$) и число рёбер m ($1 \leq m \leq 200\,000$). В следующих m строках записаны по два числа a и b , обозначающих ребро из вершины a в вершину b . Вершины пронумерованы от 1 до n . Каждая пара (a, b) встречается не более одного раза.

Формат выходных данных

В первой строке выведите n символов — i -й символ означает результат игры, если Геннадий начинает в вершине i . Во второй строке выведите n символов — i -й символ означает результат игры, если Георгий начинает в вершине i . Результат игры обозначается буквой «W», если начинающий игру игрок побеждает, «L» — если проигрывает, и «D», если игра продолжается бесконечно.

Примеры

game.in	game.out
6 7 1 2 2 1 2 3 1 4 4 1 4 5 5 6	WDLDWL DWLLWL

Замечание

Вершины 3 и 6 изначально проигрышные. В вершине 5 единственный ход ведёт в вершину 6, и игрок выигрывает. Если Георгий начинает в вершине 1, или если Геннадий начинает в вершинах 2 или 4, Геннадий может всегда пойти в вершину 1 и сделать игру бесконечной. Если Георгий начинает в вершине 4, он может пойти или в вершину 1 (что приведёт к ничьей) или в вершину 5, что приведёт к поражению. Георгий предпочитает проиграть. Аналогично, из вершины 2, он предпочитает пойти в вершину 3 и победить. Из вершины 1, Геннадий может пойти в вершину 2 и проиграть, или в вершину 4 и победить. Он предпочитает победить.

